

FiestaPivot

Team Artemis

Bailey Ivancic	z5162498
Jacob Wahib	z5164984
Estella Arabi	z5165786
Harry Tang	z5116894
Nabil Shaikh	z5163480

1. Executive summary

Within the catering industry, the currently accepted business model puts the onus on the customer to seek out parties willing to take on their catering job. This presents two problems. The first problem is a difficulty barrier is created between the advertiser and the job they are seeking to outsource, as catering businesses are not often immediately visible to inexperienced individuals/groups. The second problem that arises results from this lack of visibility, specifically small, inexperienced catering businesses or even individuals willing to put their cooking skills to use find it very difficult to find jobs they are willing to complete, without expensive advertising or other forms of exposure. FiestaPivot aims to shift this paradigm, in order to put more control in the hands of both the advertiser and bidder, in a way that is much more cost-effective and efficient. As a result, both sides of the catering equation, whether it be advertiser or bidder, will be able to reduce their overhead in terms of time taken to manage catering jobs, and use that time (and sometimes money) more effectively.

On the FiestaPivot website, a user can function as both an advertiser and a bidder. An advertiser is able to post a job they are wanting to complete, which is then available for all bidders to see, and subsequently offer their services for. When posting an ad, the advertiser includes information such as location, price, date and other job specifics to help bidders decide whether they want to offer their services or not. The advertiser is in full control of their ad, and can accept any one of the bids they receive based on the price offered, the notes the bidder leaves, or the feedback rating of the bidder from past jobs they have completed.

Bidders are able to search for ads they would like to fulfil a number of ways, to help filter all ads based on which ones would be most relevant to them. These include searching by keyword, which will search both the title and description of an ad, searching by City or State, searching using a minimum and maximum price as well as a starting date. The search feature is designed to allow bidders to be able to find jobs that match their capabilities and requirements quickly and efficiently. The interface for creating ads, searching ad and creating bids is designed to be as simple and intuitive as possible, again aiming to make the experience for both parties of the catering equation easy and

Users will have access to all the jobs they have pending or active, as well as all their bids that are either pending or accepted. This information is all available in the user dashboard, helping the user to track their jobs or their business at a quick glance.

2. Requirements

The following requirements list will be prioritised using the MoSCoW notation:

- **[Priority 1]**: 'Must Have' features. These are requirements that must be in the final version of 'FiestaPivot' MVP. Without them our project would not be able to provide its most basic functionalities and thus will be considered useless.
- **[Priority 2]**: 'Should Have' features. These are requirements that have been denoted as important, but not critical to the MVP.
- **[Priority 3]**: 'Could Have' features. These are requirements that have been deemed desirable but not crucial. These requirements would be good to add to a future release of the software, but not planned to be in the initial release
- **[Priority 4]**: 'Won't Have' features. These are requirements that the development team have denoted as the least critical. These requirements would add value to the product, but are at the lowest level of priority, and as such are unlikely to be implemented soon

Priority values were given to tasks based on how important we thought they would be to the final product. As such, priority 1 denotes items that make up the core of the application, with 2, 3, and 4 all decreasing in importance to the final deliverable.

The requirements for 'FiestaPivot' have been placed in categories, based on common functionality/action. For the following requirements, the following language will be used to represent different functionality:

*Advertiser - A registered and logged-in user who posts an advertisement on the service
Bidder - A registered and logged-in user who accepts and bids on jobs that have been posted on the system*

A logged-in user on the application is able to function as both an advertiser and bidder, these names are simply given collect requirements dealing with a similar action.

User (common)

- 1.1. A registered user is able to log into the system [Priority 1]
- 1.2 An unregistered user is able to register for the system with certain details [Priority 1]
 - 1.2.1 Username [Priority 1]
 - 1.2.2 Password [Priority 1]
 - 1.2.3 Email address [Priority 1]
 - 1.2.4 State [Priority 2]
 - 1.2.5 City [Priority 1]
 - 1.2.6 About Me [Priority 2]
 - 1.2.7 Phone Number [Priority 2]
- 1.3. A logged-in user is able to change their password [Priority 3]
- 1.4. A logged-in user can update the credentials/description of their profile [Priority 3]
- 1.5 A not-logged-in (or "guest") user will have limited access to the website [Priority 1]
 - 1.5.1 Unable to search for ads [Priority 1]
 - 1.5.2 Unable to access dashboard [Priority 1]
 - 1.5.3 Will be prompted to login upon attempting either of these things [Priority 1]

Advertisers

- 2.1. An advertiser is able to post an advertisement onto the system [Priority 1]
 - 2.1.1. An advertiser can include filterable criteria [Priority 1]
 - 2.1.1.1 Date and Time [Priority 1]
 - 2.1.1.2 Location [Priority 1]
 - 2.1.1.3 Budget [Priority 1]
 - 2.1.1.4 Alcohol provisions [Priority 1]
 - 2.1.1.5 Number of people [Priority 1]
 - 2.1.1.6 Indoors or outdoors [Priority 1]
 - 2.1.1.7 Expiration date of advertisement [Priority 1]
 - 2.1.2. An advertiser can include searchable criteria [Priority 1]
 - 2.1.2.1. Title [Priority 1]
 - 2.1.2.2 Event description [Priority 2]
- 2.2. An advertiser is able to close a job that they have previously posted [Priority 2]
 - 2.2.1. Success (bid accepted) [Priority 2]
 - 2.2.2. Expired (no bidders) [Priority 2]
 - 2.2.3. Closed (Advertisement cancelled before expiration date) [Priority 2]
- 2.3. Advertiser can see all bids on their advertisement [Priority 2]
 - 2.3.1. An advertiser is able to accept a bid from a bidder on a job they have posted [Priority 2]
 - 2.3.2. An advertiser can see the details of a bid [Priority 2]
 - 2.3.2.1. Name [Priority 2]
 - 2.3.2.2. Rating [Priority 3]
 - 2.3.2.3. Price requested [Priority 3]
 - 2.3.2.4. Other comments [Priority 3]
- 2.4. An advertiser is able to leave a feedback rating for a bidder that has previously completed a job they have posted (star rating out of 10) [Priority 3]
- 2.5. An advertiser is able to view all their posted advertisements by status [Priority 3]
 - 2.5.1. Active [Priority 3]
 - 2.5.2. Expired/closed [Priority 3]

Bidders

- 3.1. A bidder is able to bid on a job that an advertiser has posted on the system [Priority 1]
- 3.2. A bidder is able to view all their bids [Priority 1]
 - 3.2.1. Pending [Priority 2]
 - 3.2.2. Accepted [Priority 2]
 - 3.2.3. Declined [Priority 2]
 - 3.2.4 Ad Deleted [Priority 2]
- 3.3. A bidder is able to leave rating for an advertiser whose job they have completed [Priority 3]
- 3.4. A bidder can search for ads that match specified criteria [Priority 1]
 - 3.4.1. Date and Time [Priority 1]
 - 3.4.2. Location [Priority 1]
 - 3.4.3. Budget [Priority 1]
 - 3.4.4. Alcohol provisions [Priority 1]
 - 3.4.5. Number of people [Priority 1]
 - 3.4.6. Indoors or outdoors [Priority 1]
 - 3.4.7 Expiration date of advertisement [Priority 1]
- 3.5. A bidder can search for ads using keywords (found in the title and description) [Priority 2]
- 3.6. A bidder can look at details of an advertisement [Priority 2]
 - 3.6.1. Details about the poster [Priority 3]
 - 3.6.2. Description of the ad [Priority 3]
- 3.7. A bidder is able to remove a bid they have made on an active advertisement [Priority 3]

3. Use cases

UC Id	UC1
UC Name	Searching and Bidding
Actor	Bidder (user)
Requirements	2.3.2.1, 2.3.2.2, 2.3.2.3, 2.3.2.4, 3.1, 3.4.2, 3.4.3, 3.4.6, 3.5, 3.6.1, 3.6.2
UC Description	Searching for an advertisement and placing a bid on it
Pre-conditions	User has logged in and is on the landing page
Post-conditions	User has placed a bid on an event they searched for.
Interactions	<ul style="list-style-type: none"> - Bidder selects the “Indoor/Outdoor”, “Location” and “Budget” filters, types in “birthday” in the search bar, and then clicks the “Search” button. - System searches for events that meet the required filters, then opens the “Search Results” page, which shows, in list form, the advertisements in its database that meet those requirements, as well as the description and host for each event, as well as its current bid count. - Bidder scrolls through the list, selects an event, and clicks “bid”. - System issues a prompt requesting the bid price, description of their service and contact details. - Bidder fills in these fields and clicks “submit”. - System adds information to its database - System showcases that bid to the advertiser and bidder on their accounts - System adds to the bid count of the advertisement. - System will also display the bid as “Pending” on the bidder’s account.

UC ID	UC2
Requirements	2.3, 2.3.2, 3.2, 3.7
UC Name	Removing a Bid
Actor	Bidder (User)
UC Description	Removing a bid previously placed on an active advertisement
Pre-conditions	User has logged in and is on the landing page; user has already placed a bid on an advertisement
Post-Conditions	User does not have a pending bid for the advertisement they had previously bid for
Interactions	<ul style="list-style-type: none"> - User clicks “View Account” - System displays a list of their past and present (pending) bids - User clicks on the bid they want to remove and selects the “Remove Bid” option - The system removes that bid from the user’s account, the advertiser’s account, and the total bid count for the advertisement.

UC ID	UC3
Requirements	2.1, 2.1.1, 2.1.2, 2.3, 2.3.1, 2.3.2, 2.5, 2.5.1, 2.5.2, 3.1, 3.2, 3.4, 3.6
UC Name	Completing an Event
Actors	Bidder (user), Advertiser (user)
UC Description	An advertiser posts an advertisement and eventually accepts one of the bids. After the event has taken place, both advertiser and bidder are prompted by the system to write a review for the other user.
Pre-conditions	Both users are logged in and are on the landing page.

Post-Conditions	Advertiser's advertisement marked as "Closed" in their account, bidder's bid marked as "Completed" in their account.
Interactions	<ul style="list-style-type: none"> - A user creates an advertisement (filling in all required details) and clicks "Post Advertisement" - The system uploads that advertisement to the database to become available for bids. - A bidder searches for and bids on the advertisement (as in UC1) - The advertiser clicks "View Account" to see the bid. - The advertiser is pleased with the bid and clicks "Accept bid". - The system prompts the advertiser by asking "Are you sure you would like to select this bid? This will close your advertisement" - The advertiser clicks "yes". - The system changes the status of the advertisement to "Closed" on the advertiser's account - The system changes the status of the bid to "Completed" on the bidders' accounts - The system removes the advertisement from the database of searchable advertisements. - The advertiser directly contacts the bidder and the event is planned.

UC ID	UC4
Requirements	2.1, 2.2.2, 2.5, 2.5.2
UC Name	Expired Advertisement
Actor	Advertiser (user)
UC Description	A user creates an advertisement for an event and submits it but no bids are accepted and the advertisement expires at the specified expiration date.
Pre-conditions	An advertiser has created an advertisement.

Post-Conditions	The advertisement is marked as “Expired” in the advertiser’s account and is no longer in the database of searchable advertisements.
Interactions	<ul style="list-style-type: none"> - The user does not accept any of the bids for their advertisement, and it is beyond the scheduled date of the advertised event. - The system marks the advertisement as “Expired” on the user’s account page, and removes it from the database of searchable advertisements.

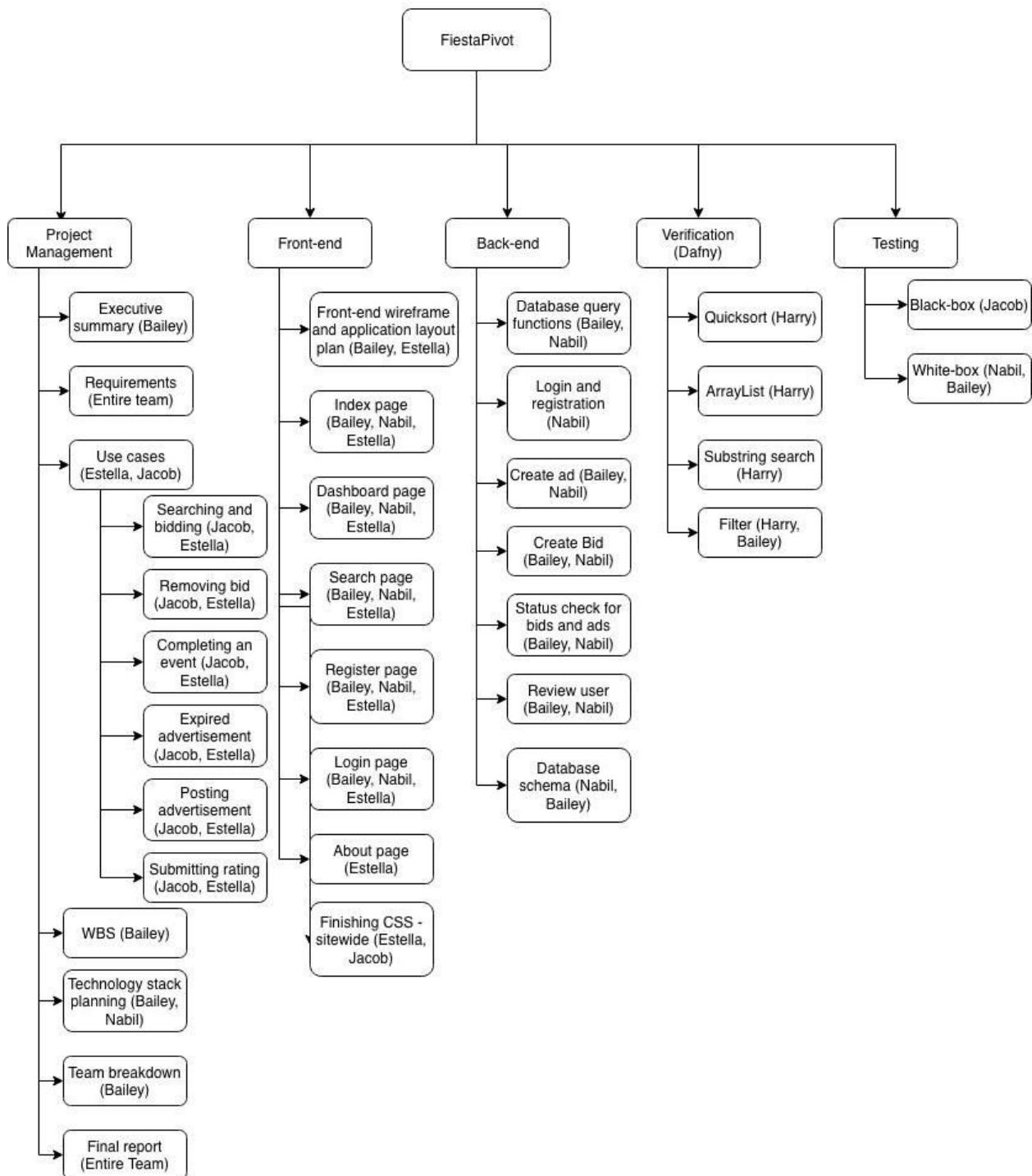
UC ID	UC5
Requirements	2.1, 2.1.1, 2.1.1.1-5, 2.1.1.7, 2.1.2, 2.1.2.1, 2.1.2.2, 2.5
UC Name	Posting an Advertisement
Actor	Advertiser (user)
UC Description	An advertiser makes and submits an advertisement to the website
Pre-conditions	User has logged in and is on the landing page
Post-Conditions	User has posted an advertisement for an event
Interactions	<ul style="list-style-type: none"> - The user clicks the “Post an Ad” button - The system redirects the user to the advertisement creation page. - The user writes a title and description for the event - The user specifies additional filters (date and time, location, budget, alcohol provisions, number of people, indoors or outdoors, expiration date of advertisement) - The user clicks “Post” - The system adds the advertisement to the database and it becomes available for other users to search and bid on.

UC ID	UC6
Requirements	3.3
UC Name	Submitting a rating - Bidder
Actor	Bidder (user)
UC Description	<p>Following the completion of an event, a bidder leaves a rating on the advertiser's account that will contribute to their "My FiestaScore" (an average of all such ratings out of 10)</p>
Pre-conditions	Advertiser has completed an event and it is passed the date and time advertised on the event, and the user has placed a bid that was accepted by the advertiser. User is also logged and on their dashboard
Post-Conditions	Both users have submitted the rating which affect the average of their personal "My FiestaScore" points
Interactions	<ul style="list-style-type: none"> - The bidder clicks the "My Bids" tab on their dashboard - The system redirects the user to this tab containing a list of their undeleted bids, including one with status "COMPLETED - PENDING REVIEW". - The bidder clicks on this ad - The system shows the user a modal, asking them to submit a "FiestaScore - Out of 10" for the advertiser of the event. - The user enters a "FiestaScore" and clicks "Submit Review" - The system adds the "FiestaScore" to the average of the advertiser's account, and redirects the user to the dashboard

UC ID	UC7
Requirements	2.4
UC Name	Submitting a rating - Advertiser
Actor	Advertiser (user)
UC Description	<p>Following the completion of an event, an advertiser leaves a rating on the caterer's (bidder's) account that will contribute to their "My FiestaScore" (an average of all such ratings out of 10)</p>
Pre-conditions	Advertiser has completed an event and it is passed the date and time advertised on the event, and the bidder has placed a bid that was accepted by the user. User is also logged and on their dashboard under the "My Ads" tab.
Post-Conditions	Both users have submitted the rating which affect the average of their personal "My FiestaScore" points
Interactions	<ul style="list-style-type: none"> - The user clicks on the completed ad that has status "COMPLETED - PENDING REVIEW" - The system shows the user a modal, asking them to submit a "FiestaScore - Out of 10" for the bidder of the event whose bid was accepted - The user enters a "FiestaScore" and clicks "Submit Review" - The system adds the "FiestaScore" to the average of the bidder's account, and redirects the user to the dashboard

4. Work breakdown structure

Work Breakdown structure is done by using a "Phased" approach.
The work has been broken down into logical components, which are grouped based on which part of the application they are contained within.



5. Implementation

Front-End (GUI)

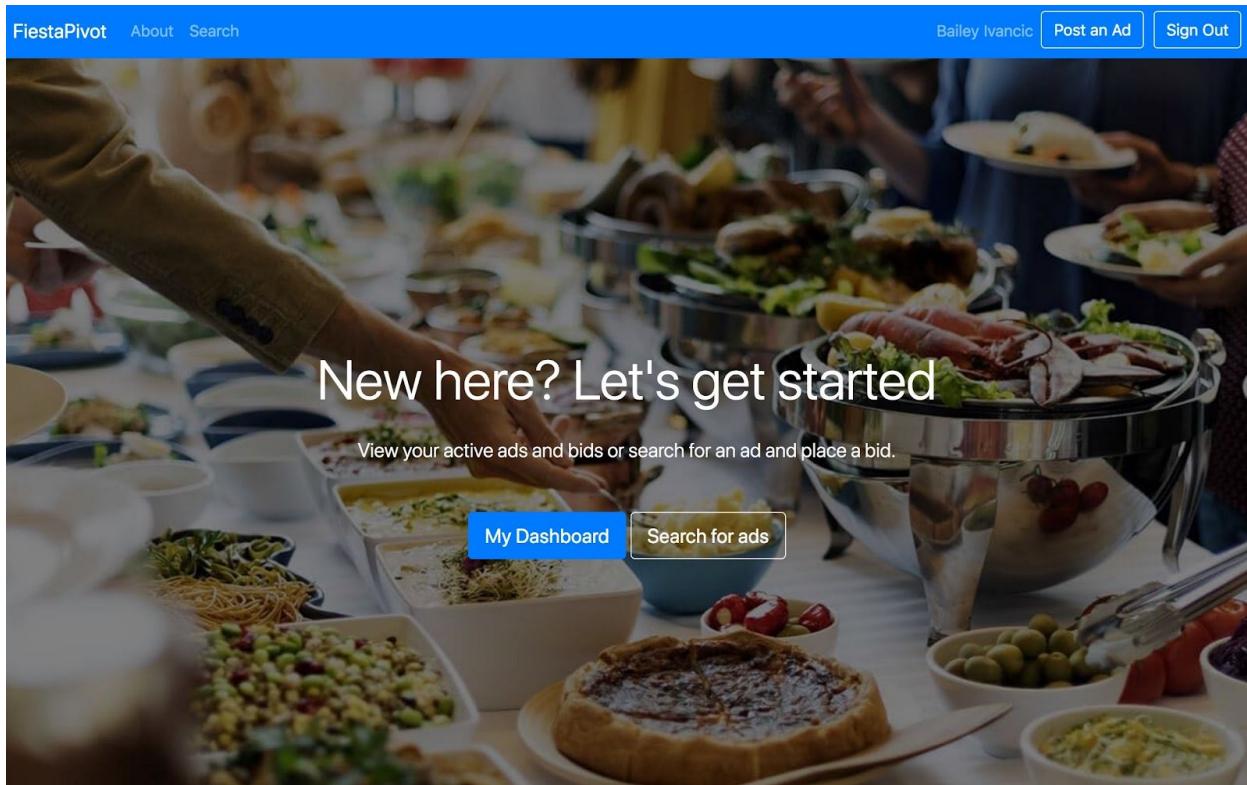
Technology and structure

HTML: HTML has been used as the elemental framework of FiestaPivot. All web pages, links and site navigation has been handled by HTML.

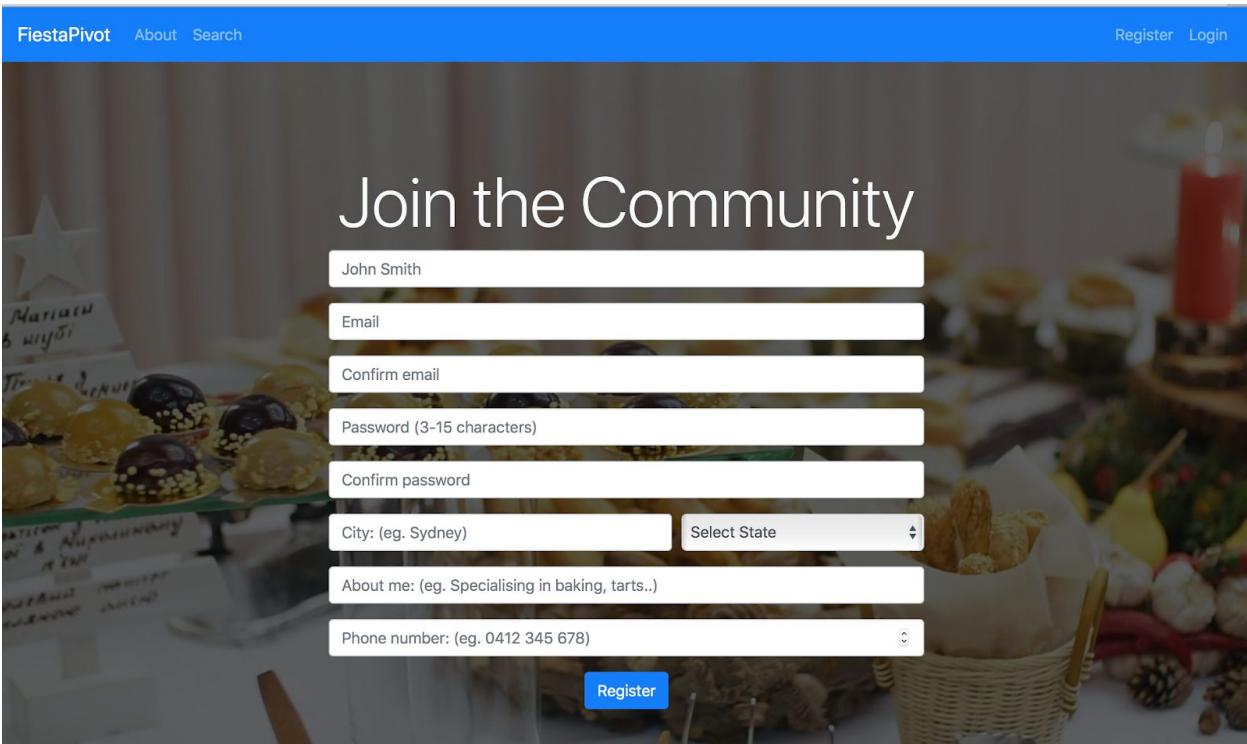
CSS: CSS has been used for visual enhancements to the base HTML on the site. The Bootstrap and W3 schools were used as a visual framework to enhance the base HTML on the site. These CSS and Javascript frameworks were combined with our own CSS, and changed to suit the UI. Javascript was used from the W3 schools as well as Bootstrap, and all other functions have been developed to suit the specific use. All work that has not been created for the project has been attributed where applicable.

Snapshots

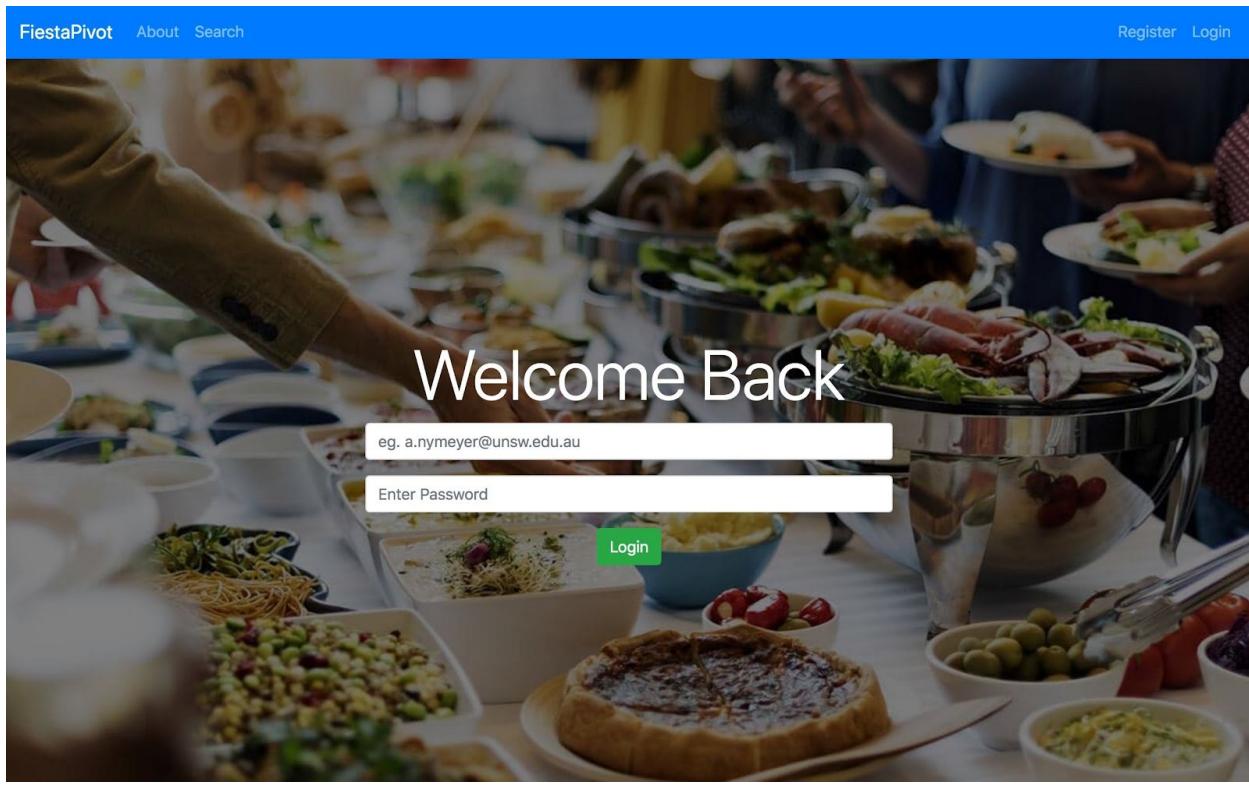
Homepage



Register



Login



User Dashboard

The image shows the user dashboard for a user named John Smith. The top navigation bar is identical to the login page, with "FiestaPivot", "About", "Search", "John Smith", "Post an Ad", and "Sign Out". The main content area features a "Welcome Back John Smith" message. Below this, there are three navigation tabs: "My Ads" (highlighted in white), "My Bids", and "My Info". A note below the tabs says "Click on \"Post an Ad\" tab to create your first ad!". The rest of the dashboard is currently blank.

My Info

FiestaPivot About Search John Smith Post an Ad Sign Out

Welcome Back John Smith

My Ads My Bids My Info

My Info

Email: another@gmail.com
City: Sydney
State: NSW
About me: New to the FiestaPivot community but excited to start using its services!
Phone number: 0490761754
Events held: 0
Bids completed: 0
My FiestaScore: 0

Change Password Edit

Editing Info

FiestaPivot About Search John Smith Post an Ad Sign Out

Welcome Back John Smith

Edit your details

John Smith
another@gmail.com
Sydney NSW
New to the FiestaPivot com
0490761754

Confirm Changes

Post an Ad

The screenshot shows the 'Create Your Ad' page for the FiestaPivot platform. At the top, there's a blue header bar with the 'FiestaPivot' logo, 'About', and 'Search' links on the left, and 'John Smith' and 'Sign Out' on the right. The main area has a white background with a dark sidebar on the left and right.

Name of event:

City: **State:**

Alcohol?

Number of People:

Price:

Description:

Date of event:
Start time of event:
End time of event:

Post

FiestaPivot is an innovative new take on the outsourcing business model - bringing all its benefits to the realm of food preparation. Find people to cater for your party, black tie function, or picnic, or offer your services to help new clients, and become the most in-demand business in town.

© 2018 Copyright Team Artemis

Viewing Self-Made Ad in Dashboard

The screenshot shows a user profile 'Yakob' at the top right. Below the header, there's a navigation bar with 'FiestaPivot', 'About', and 'Search' on the left, and 'Post an Ad' and 'Sign Out' on the right. The main content area displays a card for an ad titled 'BBQ with the Gods'. The card includes the event date '2018-11-14', a description of the event ('Event at Mount Olympus, bbq apparatus provided, would like a casual menu with burgers, ribs, etc and plenty of dessert!'), and details about the price ('Price: \$1232'), alcohol provisions ('Yes'), number of people ('50'), and status ('ACTIVE'). There are 'Delete' and 'Edit' buttons at the bottom of the card.

List of bids on ad

The screenshot shows the same user profile 'Yakob' at the top right. The main content area displays the 'BBQ with the Gods' ad card. Below the ad card, there's a section titled 'Choose a bid:' containing two bid entries. The first bid is from a user named 'another@gmail.com' with a message: 'As a seasoned Dad, I am experienced with handling a grill thanks to my own weekly Sunday BBQs. I would love to be able to cater to your event!' and a bid amount of '\$1232'. The second bid is from a user named 'yes@gmail.com' with a message: 'As a goddess myself I have great experience with what immortal beings prefer in a menu. Happy to discuss more over email if you choose my bid!' and a bid amount of '\$1200'. Both bids include the date they were posted ('2018-11-14' and '2018-03-03'). At the bottom of the page, there are 'Delete' and 'Edit' buttons.

Editing Ad

The screenshot shows a modal window titled "Edit Ad" over a dark background. The modal contains fields for event details and a description area.

Name of event: BBQ with the Gods

City: Sydney **State:** NSW

Alcohol?: Yes

Number of People: 50

Price: 1232

Description:

Event at Mount Olympus, bbq apparatus provided, would like a casual menu with burgers, ribs, etc and plenty of dessert!

Date of event: 2018-11-14 **Start time of event:** 12:30 **End time of event:** 17:00

Confirm Changes

Search page

FiestaPivot About Search Artemis, Goddess Post an Ad Sign Out

Search for ads

Please enter in criteria:

Keywords

City

State

Min. Price

Max. Price

From date:

Sort by:

Size:

Alcohol

BBQ with the Gods
Date: 2018-11-14
Event at Mount Olympus, bbq apparatus provided, would like a casual menu with burgers, ribs, etc and plenty of dessert!
Posted by: estella.arabi@gmail.com | Price: 1232 | City: Sydney State: NSW | Alcohol provisions: Yes | Number of people: 50 |

Birthday Party
Date: 2018-12-12
Looking for a caterer for my son's 10th birthday party. Superhero themed, bouncy castle and hyperactive children guaranteed. Would like for the following to be provided: cake, light snacks, and the biggest pizza your oven will make. Price negotiable, but hard budget is \$150. Thanks.
Posted by: another@gmail.com | Price: 100 | City: Canberra State: NSW | Alcohol provisions: No | Number of people: 30 |

Picnic Food Preparation
Date: 2019-01-01
Help can't cook!!! Need someone to help me prepare a picnic feast for me and my friends!!
Posted by: example@hotmail.com | Price: 50 | City: Gold Coast State: QLD | Alcohol provisions: No | Number of people: 5 |

Wedding Anniversary
Date: 2019-02-14
Wanting to surprise my wife for our wedding anniversary and looking for caterers for a private beach function. Menu is negotiable, wanting something classier that matches well with red wine.
Posted by: another@gmail.com | Price: 200 | City: Canberra State: NSW | Alcohol provisions: Yes | Number of people: 15 |

Private Function
Date: 2019-03-03
Small bonfire gathering on Henley beach, would like caterers to wander the area with finger food such as hors d'oeuvres and champagne.

Search page (with filtering)

FiestaPivot About Search Mary Sue Post an Ad Sign Out

Search for ads

Please enter in criteria:

Keywords

City

State

Min. Price

Max. Price

From date:

Sort by:

Size:

Birthday Party
Date: 2018-12-12
Looking for a caterer for my son's 10th birthday party. Superhero themed, bouncy castle and hyperactive children guaranteed. Would like for the following to be provided: cake, light snacks, and the biggest pizza your oven will make. Price negotiable, but hard budget is \$150. Thanks.
Posted by: another@gmail.com | Price: 100 | City: Canberra State: NSW | Alcohol provisions: No | Number of people: 30 |

Viewing Ad from Search Page

The screenshot shows the FiestaPivot search interface. On the left, there is a sidebar titled "Search for ads" with various filters: Keywords (Type in keywords), City (E.g. Sydney), State (Choose...), Min. Price, Max. Price, From date:, Sort by: Date: Soonest first, Size: None, and Alcohol provisions: Both. On the right, a detailed view of an advertisement titled "BBQ with the Gods" is displayed. The ad is for a "Birthday Party" posted by another@gmail.com on 2018-12-12 at Canberra, NSW, with a price of 100. It includes sections for "Alcohol provisions: No", "Number of people: 30", and "Job description: Looking for a caterer for my son's 10th birthday party. Superhero themed, bouncy castle and hyperactive children guaranteed. Would like for the following to be provided: cake, light snacks, and the biggest pizza your oven will make. Price negotiable, but hard budget is \$150. Thanks." A "Place bid" button is visible at the bottom of this panel.

Placing Bid

The screenshot shows the FiestaPivot search interface with a bid placement dialog overlaid. The dialog is titled "BBQ with the Gods" and says "Bidding on: Birthday Party". It contains fields for "Original price: 100" and "Your price: 100". Below these is a "Additional messages/comments to poster:" field with "Notes to poster" entered. A "Place bid" button is at the bottom of the dialog. In the background, the search results for "BBQ with the Gods" are visible, showing other ads for "Wedding Anniversary" and "Private Function". The "Wedding Anniversary" ad is for a date of 2019-02-14, wanting to surprise a wife for a private beach function with a menu negotiable and matching red wine. The "Private Function" ad is for a date of 2019-03-03, a small bonfire gathering on Henley beach with finger food and champagne.

Viewing Bids (from Dashboard)

The screenshot shows a user profile at the top right: John Smith, Post an Ad, Sign Out. Below the header, a modal window is open with the title "BBQ with the Gods". Inside the modal, the following information is displayed:

- Bid amount: \$1232
- Message: As a seasoned Dad, I am experienced with handling a grill thanks to my own weekly Sunday BBQs. I would love to be able to cater to your event!
- Original Price: \$1232 Status: PENDING

At the bottom of the modal are two buttons: "Delete" and "Edit".

Editing Bids

The screenshot shows a user profile at the top right: John Smith, Post an Ad, Sign Out. Below the header, a modal window is open with the title "Edit bid for: BBQ with the Gods". Inside the modal, the following fields are visible:

- Original price: \$1232
- Your price: 1232
- Additional messages/comments to poster:
As a seasoned Dad, I am experienced with handling a grill thanks to my own weekly Sunday BBQs. I would love to be able to cater to your event!

At the bottom of the modal is a "Confirm Changes" button.

Choosing/Accepting Bid

The screenshot shows a modal window for choosing a bid for an event titled "BBQ with the Gods". The event details are as follows:

- Title:** BBQ with the Gods
- Date:** Event date: 2018-11-14
- Description:** Event at Mount Olympus, bbq apparatus provided, would like a casual menu with burgers, ribs, etc and plenty of dessert!
- Price:** \$1232 | Alcohol provisions: Yes | Number of people: 50 | Status: ACTIVE

Bid Options:

- Bid for BBQ with the Gods** (Message: As a seasoned Dad, I am experienced with handling a grill thanks to my own weekly barbecues. I would be happy to be able to cater to your event! Bid amount: \$1232 | Posted by user: another@gmail.com) - A blue "Choose Bid" button is present.
- Bid for BBQ with the Gods** (Message: As a goddess myself I have great experience with what immortal beings prefer in a menu. Happy to discuss more over email if you choose my bid! Bid amount: \$1200 | Posted by user: yes@gmail.com)

Buttons: Delete (grey), Edit (blue).

The screenshot shows the same modal window after a bid has been accepted. The status is now "PROGRESS". The bid details remain the same as in the previous screenshot.

Status: Accepted

Bid Options:

- Bid for BBQ with the Gods** (Message: As a seasoned Dad, I am experienced with handling a grill thanks to my own weekly barbecues. I would be happy to be able to cater to your event! Bid amount: \$1232 | Posted by user: another@gmail.com)
- Bid for BBQ with the Gods** (Message: As a goddess myself I have great experience with what immortal beings prefer in a menu. Happy to discuss more over email if you choose my bid! Bid amount: \$1200 | Posted by user: yes@gmail.com) - This option is now labeled "Declined".

Buttons: Delete (grey), Edit (blue).

Reviewing users after eventAd Status: COMPLETED - PENDING REVIEW and notification to “Review Ad”

Welcome Back Yakob

[My Ads](#) [My Bids](#) [My Info](#) [Review Ad](#)

Testing Event Event date: 2018-10-25
 Test Description
 Price: \$100 | Alcohol provisions: No | Number of people: 12 | Status: COMPLETED - PENDING REVIEW |

BBQ with the Gods Event date: 2018-11-14
 Event at Mount Olympus, bbq apparatus provided, would like a casual menu with burgers, ribs, etc and plenty of dessert!

Submit Review - Advertiser

Welcome Back Yakob

[My Ads](#) [My Bids](#) [My Info](#) [Review Ad](#)

Testing Event Event date: 2018-10-25 X
 Test Description
 Price: \$100 | Alcohol provisions: No | Number of people: 12 | Status: COMPLETED - PENDING REVIEW |

We hope that you had a great time hosting your event with FiestaPivot. Please review your experience with this bidder:
 FiestaScore (Out of 10):
 Submit Review

Private Function Event date: 2019-03-03
 Small bonfire gathering on Henley beach, would like caterers to wander the area with finger food such as hors d'oeuvres and champagne.
 Price: \$300 | Alcohol provisions: Yes | Number of people: 25 | Status: ACTIVE |

Bid Status: COMPLETED - PENDING REVIEW and message to “Review Ad”

[My Ads](#) [My Bids](#) [My Info](#)

BBQ with the Gods
 Message: As a goddess myself I have great experience with what immortal beings prefer in a menu. Happy to discuss more over email if you choose my bid!
 Bid amount: \$1200 | Original Price: \$1232 | Status: DECLINED | Date posted: 2018-10-26 |

[Review Ad](#)
Testing Event
 Message: Yes will cater
 Bid amount: \$100 | Original Price: \$100 | Status: COMPLETED - PENDING REVIEW | Date posted: 2018-10-26 |

[Submit Review - Bidder](#)

Welcome! We hope that you had a great time hosting your event with FiestaPivot. Please review and share your experience with this advertiser:

FiestaScore (Out of 10):

My Ads

BBQ with friends

Message: I will choose my bid!

Bid amount: \$1200 | Original Price: \$1232 | Status: DECLINED | Date posted: 2018-10-26 |

Review Ad

Testing Event

Message: Yes will cater

[About Page](#)

FiestaPivot About Search John Smith [Post an Ad](#) [Sign Out](#)

What do we do?

We help people help people.

FiestaPivot aims to bring outsourcing and all its benefits to the realm of food preparation. Sign up as an individual and find people to cater for your party, black tie function, or picnic. Alternatively, offer your services as a business to help new clients enjoy good food. Any other inspiring words and calls to action will go here, except for now we have just left this rambling here instead because coming up with something that sounds nice will probably take us the same amount of time as if we were to fix the front-end of the rest of the website!

Our Founder

Artemis, goddess of the hunt, animals and the wild, is the fearsome leader who inspires our team everyday to strive to better people's lives. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed in fermentum diam. Suspendisse nisl orci, tristique sit amet ligula vitae, porta mattis justo. Donec nec tellus sit amet odio efficitur tincidunt at ac nulla. Fusce at mi ac nunc iaculis dictum. Integer posuere vitae quam in dapibus. Nam varius quam ipsum, et sodales sem mattis vitae. Aliquam laoreet nec nibh eget vestibulum. Aliquam ullamcorper lectus eget tincidunt faucibus. Donec imperdiet luctus scelerisque.



Our Team



Bailey Ivancic



Nabil Shaikh



Jacob Wahib



Harry Tang



Estella Arabi

FiestaPivot is an innovative new take on the outsourcing business model - bringing all its benefits to the realm of food preparation. Find people to cater for your party, black tie function, or picnic, or offer your services to help new clients, and become the most in-demand business in town.

© 2018 Copyright Team Artemis

Back-End

Technology and structure

Python

Python was used as a powerful backend language to host the website and provide main functionality. Along with Flask and flask_login this became a effective yet simplistic tool that aided in fast development and flexible adaptation. Python has been used as the main logical tool in our webapp, making communication between the database, server, client and backend itself easier.

We translated all our dafny verified functions and classes to python, such that we could use them in our final product in the backend. Functions that have helped us sort, filter, search and other classes that have allowed us to use our own verified data types such as an array list.

Jinja2

Jinja has been used lightly throughout the development process to add extra dynamic to our html templates. Jinja and Flask work together really well in communicating between the backend and the frontend, having an impact on aesthetic and major functionality. We used Jinja2 extensively as a templating tool for displaying our database results. Jinja2 allows for dynamic and adaptive displaying of data, and combines seamlessly with our flask back-end to make our interface flexible in regards to data handling and display.

JavaScript

Javascript has been used throughout the front-end in order to manipulate the dom, change which results are displayed, and for the operation of the modals. Javascript is also used for some error checking of user inputs, as well as onclick events for certain DOM elements such as displaying and hiding modals, sending requests. Javascript has been used to enhance the user experience, aid in interaction between the user and the site and add more dynamic to our site.

Our Dafny functions for isSubstring() and isPrefix() have been translated into Javascript functions, and are used on the front end within our filtering function. The user is able to search for ads using keywords and strings, and our substring function is responsible for checking if these keywords are contained within the ad title or description. If true, then the results are hidden from the user. This is combined with other javascript elements checking other criteria to complete the filtering function used for displaying and searching for ads.

Flask

Flask has been combined together with Python and Jinja2 to create a dynamic webapp. Flask's routing system works together with its inbuilt login manager to not only make interaction between different web pages easier, but also to limit the access for non-logged in users. Flask has been an effective tool to communicate to and from the backend, parsing data from the frontend using forms and back again through the use of jinja to insert information into our web pages. Flask, built on top of python, became a simplistic tool to host and aid in agile development throughout the project.

Sqlite3

SQLite3 has been used as the database for the project. All data, including accounts, ads and bids, is stored in our db. Data is queried in the back end, formatted and then sent to the front end to be displayed. Python serves as the language to communicate between sql commands and formatting data to then be transported to the frontend.

Verification

Our main functions that are critical to the application have been verified in Dafny. These functions are BubbleSort, QuickSort, isSubstring, isPrefix and an ArrayList data type. Verification in Dafny presented us with a challenge, since our back-end for the application was written in Python, so there was obviously going to be some translation required. We verified an ArrayList datatype in Dafny, which we planned on using within Python to attempt to keep it as close as possible to the Dafny code.

After talking with our mentor and Albert, however, we decided that some abstraction was acceptable in the code, and decided to simply translate the code into Python, using the default Python ArrayList. As such, we converted these functions into Python manually, with the exception of isSubstring and isPrefix.

IsSubstring and isPrefix were instead converted into Javascript functions, as these functions were used on our front end, inside our Javascript filtering function. We attempted to verify the filtering function, but had little success.

The conversion to Python and Javascript was done in a way that attempted to copy the logic of the program identically, despite the differing syntax and data structures used. As said previously, we are assuming some level of abstraction with these algorithm conversions since we are not explicitly using a converted Dafny ArrayList datatype within our Dafny functions.

We verified 3 main methods and our own custom ArrayList datatype using Dafny. Initially, we verified BubbleSort as we were generally new to Dafny and something which had good documentation and simpler invariants made it a lot more easy to verify. As we progressed further, we realised that BubbleSort is a disaster sort in terms of performance so we verified QuickSort in order to replace BubbleSort. All methods and datatypes are independent of one another.

BubbleSort

As a quick overview, our BubbleSort takes an array of integers (array<int>) and sorts them in-situ and in ascending order. When analysing BubbleSort, we noticed a lot of patterns across the different states. For example, the number of iterations is equal to the number of items that are sorted, which gives rise to the postcondition:

```
invariant forall j :: i <= j < a.Length ==> a[i] <= a[j];
```

Furthermore, from our implementation, all of the smaller elements are swapped to the left-side of the array, if you partition the array at the array's current upper bound, that is, the index of the end of sorted partition of the array, we can guarantee that all of the elements on the left side of the array are smaller than the right side of the array. These two invariants were the key to verifying BubbleSort. This was then tested using a Test method which verifies correctly.

```
invariant forall j,k :: 0 <= j < upperBound && upperBound <= k < a.Length ==> a[j] <= a[k];
```

Quicksort

Quicksort like BubbleSort takes in an array of integers (array<int>) and sorts its elements in-situ and in ascending order. Quicksort was proven to be incredibly difficult to verify. This can be explained by two issues, proving loop termination and ensuring that some methods and loop termination shared the same postconditions.

Partition has a postcondition where the all of the elements left of the pivot are less than the pivot and all of the elements on the right are greater than or equal to the pivot.

```
invariant forall i | left <= i < p :: (a[i] < a[p])
invariant forall i | p < i < k + 1 :: (a[p] <= a[i])
```

This method was verified completely with Dafny.

Searching

Our method isSubstring is our core method for searching. It takes a needle called token and haystack called str and iteratively compares the token to the haystack. A recursive method was not chosen as proving loop termination in Dafny can be considerably more complex and difficult. We formed a predicate called hasSubstringWithOffset which checks whether a token at a specific offset matches another string. We use the Dafny's prefix operator “ \leq ” as the core of

our substring search. This operator checks whether the token is a prefix of a string. This made it considerably more simple as we did not need to verify another method `isPrefix`.

The invariants for substring searching were considerably more difficult. One core invariant was that if a substring did exist, then the length of the token + the index of the offset would be less than or equal to the length of the haystack string and that at that offset, the prefix relationship would hold true.

```
invariant r ==> (exists n : int :: (n >= 0) && (n + |token| <= |str|) && (token <= str[n..]))
```

ArrayList

Our custom ArrayList data type was verified in order to think more about the big picture. This proved to be our largest component that was verified in Dafny. While it is not used due to time-constraints, it has been fully translated into Python and can be imported as a library for our backend in the future. For the final deliverable, we have used our verified Dafny functions with the inbuilt Python array, abstracting the datatype from the algorithm and just verifying the logic.

The postcondition `fresh(repr - {this})` concludes that the newly allocated variable is disjoint from the previous set of objects. This data structure and its invariants were heavily inspired by K. Rustan M. Leino's research papers as shown in Section 1.5 of [\[1\]](#).

While most invariants were less complex, the difficulty in verifying this data structure can be attributed to the numerous methods connected to one another in order for this data structure to correctly verify.

In the translated version of ArrayList, the capacity variable was abstracted as the native Python list is not designed to allocate fixed size arrays.

Filter

Our filter method did not verify correctly. Filter was designed to be a higher order function which accepts an array and predicate and returns a fresh array containing the elements in a where the predicate holds true. The introduction of co-inductive datatypes and ghost methods did not help verify this method so it was not included in our implementation.

6. Evaluation

Black-box Testing

Test Case #	Test Case Description	Step #	Test Data	Steps to Perform	Expected Result	Actual Result	Comments
1	Register a new account - valid inputs RQTS 1.2, 1.2.1-7	1	Test	Enter Username	Username appears as it is typed into the field	Username appears as it is typed into the field	Behaves as expected
		2	<u>test@test</u>	Enter Email	Email appears as it is typed into the field	Email appears as it is typed into the field	Behaves as expected
		3	<u>test@test</u>	Confirm Email	Email appears as it is typed into the field	Email appears as it is typed into the field	Behaves as expected
		4	1234	Enter Password	Password characters appear as dots as they are typed into the field	Password characters appear as dots as they are typed into the field	Behaves as expected
		5	1234	Confirm Password	Password characters appear as dots as they are typed into the field	Password characters appear as dots as they are typed into the field	Behaves as expected
		6	test	Enter City	City appears as it is typed into the field	City appears as it is typed into the field	Behaves as expected
		7	NSW	Select State	State is selected from drop-down menu	State is selected from drop-down menu	Behaves as expected
		8	I am a test	Enter About Me information	Information appears as it is typed into the field	Information appears as it is typed into the field	Behaves as expected
		9	00000000000	Enter Phone Number	Phone Number appears as it is typed into the field. Any non-digit characters are rejected	Phone Number appears as it is typed into the field. Any non-digit characters are rejected	Behaves as expected
		10		Click Register	User is signed in and redirected to their new dashboard	User is signed in and redirected to their new dashboard	Behaves as expected

2	Log out of account, ascertain guest status RQTS 1.5	1		Click Sign Out	User is logged out and redirected to the login page	User is logged out and redirected to the login page	Behaves as expected
		2		Click Search	User is redirected to login page	User is redirected to login page	Behaves as expected

3	Log into Account - correct data RQTS 1.1	1		Click Login (navbar)	User is redirected to login page	User is redirected to login page	Behaves as expected
		2	test@test	Enter Email	Email appears as it is typed into the field	Email appears as it is typed into the field	Behaves as expected
		3	1234	Enter Password	Password characters appear as dots as they are typed into the field	Password characters appear as dots as they are typed into the field	Behaves as expected
		4		Click Login (button)	User is redirected to their account's dashboard	User is redirected to their account's dashboard	Behaves as expected

		1		Click Post An Ad	User is redirected to the post-ad page	User is redirected to the post-ad page	Behaves as expected
4	Posting ad RQTS 2.1, 2.1.1, 2.1.1.1 - 2.1.1.5, 2.1.1.7, 2.1.2, 2.1.2.1, 2.1.2.2, 2.5	2	Testing Event	Enter Name of Event	Name appears as it is typed into the field	Name appears as it is typed into the field	Behaves as expected
		3	test1	Enter City	City appears as it is typed into the field	City appears as it is typed into the field	Behaves as expected
		4	NSW	Select State	State is selected from drop-down menu	State is selected from drop-down menu	Behaves as expected
		5	Yes	Select Alcohol	Alcohol is selected from drop-down menu	Alcohol is selected from drop-down menu	Behaves as expected
		6	10	Enter Number of People	Number appears as it is typed into the field. Any non-digit characters are rejected	Number appears as it is typed into the field. Any non-digit characters are rejected	Behaves as expected
		7	100	Enter Price	Price appears as it is typed into the field. Any non-digit characters are rejected	Price appears as it is typed into the field. Any non-digit characters are rejected	Behaves as expected
		8	An event to test	Enter Description of Event	Description appears as it is typed into the field	Description appears as it is typed into the field	Behaves as expected
		9	1/1/19	Select Date	Date appears in the field, being automatically filled between the '/' for day/month/year	Date appears in the field, being automatically filled between the '/' for day/month/year	Behaves as expected
		10	6:00am	Select Start Time	Time appears in the field, being automatically filled between the ':' for hh:mm with the choice for am/pm on the end	Date appears in the field, being automatically filled between the '/' for day/month/year	Behaves as expected
		11	6:00pm	Select End Time	Time appears in the field, being automatically filled between the ':' for hh:mm with the choice for am/pm on the end	Time appears in the field, being automatically filled between the ':' for hh:mm with the choice for am/pm on the end	Behaves as expected
		12		Click Post	User is redirected to their account's dashboard, where the newly posted ad may be seen under "My Ads"	User is redirected to their account's dashboard, where the newly posted ad may be seen under "My Ads"	Behaves as expected

5	Search for Ad and Bid RQTS 3.1, 3.2, 3.2.1, 3.2.2, 3.4, 3.4.1-7, 3.5, 3.6, 3.6.2	1	Click Search (navbar)	User is redirected to Search page	User is redirected to Search page	Behaves as expected
		2	Ice Cream	Enter Keywords	Keywords appear as they are typed into the field	Keywords appear as they are typed into the field
		3	Sydney	Enter City	City appears as it is typed into the field	City appears as it is typed into the field
		4	NSW	Select State	State is selected from drop-down menu	State is selected from drop-down menu
		5	100	Enter Min. Price	Min. Price appears as it is typed into the field	Min. Price appears as it is typed into the field
		6	200	Enter Max. Price	Max. Price appears as it is typed into the field	Max. Price appears as it is typed into the field
		7	23/10/18	Select From Date	Date appears in the field, being automatically filled between the '/' for day/month/year	Date appears in the field, being automatically filled between the '/' for day/month/year
		8	Soonest First	Select Sort By	Results are sorted by their dates from soonest	Results are sorted by their dates from soonest
		9	11-20.	Select Size	Size appears as it is typed into the field. Any non-digit characters are rejected	Size appears as it is typed into the field. Any non-digit characters are rejected
		10	No	Select Alcohol Provisions	"No" appears in the field	"No" appears in the field
		11		Click Refresh	Results are filtered and sorted according to the specification	Results are filtered and sorted according to the specification
		12	"Ice cream"	Click selected event to bid on	Modal appears depicting more information of the event and the option to place bid	Modal appears depicting more information of the event and the option to select bid
		13		Click Place Bid	Another modal appears prompting the user to choose a price and an optional description of the bid	Another modal appears prompting the user to choose a price and an optional description of the bid
		14	100	Enter Your Price	Price appears as it is typed into the field. Any non-digit characters are rejected	Price appears as it is typed into the field. Any non-digit characters are rejected
		15	"Ice cream"	Add note to Poster	Note appears it is typed into the field	Note appears it is typed into the field
		16		Click Place Bid	Modal closes, leaving the user on the search results page. The ad they bid on is no longer visible in the search results	Modal closes, leaving the user on the search results page. The ad they bid on is no longer visible in the search results
		17		Click [username] (navbar)	User is redirected to their account's dashboard	User is redirected to their account's dashboard
		18		Click "My Bids"	User sees the bid they just posted on the respective ad with status "PENDING"	User sees the bid they just posted on the respective ad with status "PENDING"

6	Select Ad RQTS 2.2	1		Click [username] (navbar)	User is redirected to their account's dashboard	User is redirected to their account's dashboard	Behaves as expected
		2	"Testing Event"	Click on posted ad that has one or more bids	Modal appears depicting more information of the event and the list of current bids on the ad	Modal appears depicting more information of the event and the list of current bids on the ad	Behaves as expected
		3	"Bid for Testing Event"	Hover mouse over selected bid	Option to Choose Bid appears	Option to Choose Bid appears	Behaves as expected
		4		Click "Choose Bid"	Bid is selected and modal is closed. Status of ad is now "PROGRESS"	Bid is selected and modal is closed. Status of ad is now "PROGRESS"	Behaves as expected
		5		Click on that same ad	Selected bid now says "Accepted"	Selected bid now says "Accepted"	Behaves as expected
7	Delete Ads RQTS 2.2.1-3, 2.5, 2.5.1-2, 3.2, 3.2.1-4	1		Click [username] (navbar)	User is redirected to their account's dashboard	User is redirected to their account's dashboard	Behaves as expected
		2	"Test 1"	Select ad with accepted bid to delete	Modal appears, depicting the list of bids as well as the selected bid	Modal appears, depicting the list of bids as well as the selected bid	Behaves as expected
		3		Click "Delete"	Modal is closed, the user is back on their "My Ads" tab in their dashboard	Modal is closed, the user is back on their "My Ads" tab in their dashboard	Behaves as expected
		4	"Test 2"	Select ad with pending bid to delete	Modal appears, depicting the list of bids	Modal appears, depicting the list of bids	Behaves as expected
		5		Click "Delete"	Modal is closed, the user is back on their "My Ads" tab in their dashboard	Modal is closed, the user is back on their "My Ads" tab in their dashboard	Behaves as expected
		6		Open bidders' dashboards (who had bids on the deleted ads) dashboard (using same processes as tests 2 and 3)	User is at their dashboard	User is at their dashboard	Behaves as expected
		7		Click "My Bids"	The bid of the user whose bid was accepted will have its status as "ACCEPTED", the bids of the other uses on this ad will have their statuses set to "DECLINED", and the bids on the deleted ad with no chosen bids will have their statuses as "AD DELETED"	The bid of the user whose bid was accepted will have its status as "ACCEPTED", the bids of the other uses on this ad will have their statuses set to "DECLINED", and the bids on the deleted ad with no chosen bids will have their statuses as "AD DELETED"	Behaves as expected

8	Delete Bid RQTS 3.7	1		Click [username] (navbar)	User is redirected to their account's dashboard	User is redirected to their account's dashboard	Behaves as expected
		2		Click "My Bids"	User sees the list of their bids	User sees the list of their bids	Behaves as expected
		3	"Bid for First Bid"	Click selected Bid to delete	Modal appears depicting details of the bid	Modal appears depicting details of the bid	Behaves as expected
		4		Click "Delete"	Modal disappears and the bid is no longer in the list, which the user sees after being redirected to the "My Bids" tab	Modal disappears and the bid is no longer in the list, which the user sees after being redirected to the "My Bids" tab	Behaves as expected
		5	test@gmail.com	Open dashboard of account which contained the ad	User is on their dashboard viewing the list of ads	User is on their dashboard viewing the list of ads	Behaves as expected
		6	"First Bid"	Click on ad which had its bid deleted	Modal appears containing list of bids on that ad, excluding the one that was just deleted	Modal appears containing list of bids on that ad, excluding the one that was just deleted	Behaves as expected

9	Post Ratings RQTS 2.4, 3.3	1	test@gmail.com	Log into account containing an ad with a completed bid that is past its expiry date	User is on their dashboard viewing the list of ads, one of which meets the specified criteria and has status "COMPLETED - PENDING REVIEW", with an additional notification above the ad to "Review Ad"	User is on their dashboard viewing the list of ads, one of which meets the specified criteria and has status "COMPLETED - PENDING REVIEW", with an additional notification above the ad to "Review Ad"	Refer to Test #3, behaves as expected
		2		Click on this ad	Modal appears prompting the user to enter a "FiestaScore - Out of 10"	Modal appears prompting the user to enter a "FiestaScore - Out of 10"	Behaves as expected
		3	"6"	Enter Rating (out of 10)	Rating appears as it is entered	Rating appears as it is entered	Behaves as expected
		4		Click "Submit Review"	User is redirected to their account's dashboard, on the "My Ads" tab	User is redirected to their account's dashboard, on the "My Ads" tab	Behaves as expected
		5	bidder@bidder	Log into account that had bidden on this completed ad	User is on their dashboard on the "My Ads" tab	User is on their dashboard on the "My Ads" tab	Refer to Test #3. Behaves as expected
		6		Click "My Bids"	User sees a list of their bids, including their bid on the completed ad which has status "COMPLETED - PENDING REVIEW"	User sees a list of their bids, including their bid on the completed ad which has status "COMPLETED - PENDING REVIEW"	Behaves as expected
		7		Click on bid on the completed ad	Modal appears, prompting the user to enter a "FiestaScore - Out of 10"	Modal appears, prompting the user to enter a "FiestaScore - Out of 10"	Behaves as expected
		8		Enter rating (out of 10)	Rating appears as it is entered	Rating appears as it is entered	Behaves as expected
		9		Click "Submit Review"	User is redirected to their account's dashboard, on the "My Ads" tab	User is redirected to their account's dashboard, on the "My Ads" tab	Behaves as expected

Requirements Status Table

Requirement	Implemented: ✓ = implemented ✗ = not implemented ○ = partially implemented	Reflection
1.1 - Login	✓	Our code would be inoperable without login/logout functionality, shown by its Priority 1 status, so it was one of the earlier features we implemented
1.2 - Register	✓	Register is a crucial Priority 1 component of FiestaPivot, as we the feature to have multiple accounts to permit logging in
1.2.1-7 - Register with Username, Password, Email Address, State, City, "About Me", and Phone Number	✓	We managed to include all of the desired data for Register - both Priority 1 and Priority 2-marked fields
1.3 - Change Password	✓	Changing Password was successfully implemented, though it was one of the last things completed as appropriately allocated by its Priority 3 status.
1.4 - Edit Profile	✓	Editing Profile was a Priority 3-level feature, which we managed to implement successfully in the

		later stages of development
1.5, 1.5.1-3 - Limited guest access: can't access dashboard, search for ads, or post an ad	✓	We were late to name this as a priority, but we ensured early on that any attempts to access the aforementioned pages prior to being logged in would redirect the user to the login page.
2.1 - Post ad	✓	Post ad is one of the basic requirements of the system, so this was given Priority 1 and implemented quite early in development
2.1.1, 2.1.1.1-5 - Include ad information: Date/Time, Location, Budget, Alcohol Provisions and Number of people	✓	Some of these fields were perhaps not worthy of their Priority 1 status as we could achieve basic functionality without them, but they were all implemented together nonetheless
2.1.1.6-7 - Extra ad information: Indoors/Outdoors, Expiration date of ad	X	These fields, in spite of their Priority 1 status, we did not implement, and their high priority was inappropriately allocated to them as opposed to a Priority 2 or 3
2.1.2, 2.1.2.1-2 - Searchable criteria: Title of event and Event description	✓	These fields were crucial in testing our search functionality, so we implemented them early on in accordance with

		their Priority 1 status.
2.2 - Closing ad	✓	An ad may be closed by the user deleting them or having their status changed to "EXPIRED" once the advertised event's date was reached. This feature is one of our more prominent Priority 2 features. It is worth noting that our idea for how this would function from the time of making the requirements and implementing it change, since we imagined a more diverse set of cases to cater for in the requirements.
2.2.1 - Closing ad with: no bidders after deadline	✓	An ad that meets this criteria will have its status set to "EXPIRED" and it will be the choice of the user to delete this ad or not
2.2.2-3 - Closing ad with: accepted bid, no accepted bid	✓	Both of these cases are handled by the option to delete an ad, and the statuses of the bids in bidders' accounts are changed. Not exactly how the requirements indicate it would behave, but this functionality is catered to nonetheless.
2.3 - Advertiser can view bids	✓	Including this on a user dashboard was

		fairly important and this requirement may well have been upgraded to a Priority 1, but it was rapidly implemented nonetheless.
2.3.1 - Advertiser can accept bid	✓	This feature may also have been upgraded to Priority 1 since it is quite crucial to the system, but its Priority 2 status made us rather slothful in implementing it.
2.3.2 - Advertiser can see details of bid	✓	We chose to include in the same modal that the bid is chosen from, so both Priority 2 requirements are handled almost simultaneously
2.3.2.1-4 - Advertiser can view: name, rating, bid price, and bidder's comments	✓	Despite some of these field being Priority 3, these were all implemented together, save for rating, which was developed much later in accordance with its Priority 3 status.
2.4 - Advertiser can leave feedback for bidder	○	Ratings - which we call "FiestaScore" - was one of the last things we implemented. It was not as complete as we envisaged with a full feedback + rating system, but we have a working feedback

		option (Integer value only, without message) which will modify the overall "My FiestaScore" of a user, where "My FiestaScore" is the average of all ratings given to the user. We anticipated Ratings may not have made it into the system, which is why it was allocated Priority 3.
2.5 - Advertiser can view their posted advertisements	✓	This requirement should probably have been upped to Priority 2, as it was achieved quite early and is a heavily relied-upon function since we built on it to implement the "view bid" functionality, but it too was completed earlier on in development
2.5.1-2 - Advertiser can view the status of their advertisements: Active, Expired or Completed	✓	We felt this was prioritised and implemented appropriately
3.1 - User can bid on an ad	✓	This is an elemental function of our system and was implemented earlier on in conjunction with its Priority 1 status.
3.2, 3.2.1-4 - User can view all their bids, and check the status of those bids: Pending, Accepted, Declined, or Ad Deleted	✓	Viewing bids for the purpose of editing them, deleting them, or just monitoring was quite important to our system, and these Priority 1-2

		requirements were thus implemented
3.3 - A bidder can leave a rating for the advertiser who accepted their bid	○	The rating for bidders is also a contributor to a user's "My FiestaScore", meaning the rating system is shared between advertiser ratings and caterer-ratings. Ideally they would be separated with an individual rating for each, however since these features were Priority 3 they were saved till the very end and were not implemented to the fullest
3.4 - User can search for bids	✓	Searching for bids was also quite important in our system, so it was given Priority 1 and implemented quickly
3.4.1-5 - User can filter search results by certain criteria: Date/Time, Location, Budget, Alcohol Provisions and Number of people	✓	Filtering was considered integral to our system since we expected to verify its functionality with Dafny so it was given Priority 1 and implemented quickly after search
3.4.6-7 - User can filter by: Indoors/Outdoors or Ad's Expiration date	✗	These fields did not make into the system and thus were not implemented, despite their Priority 1 status which we now acknowledge was inappropriate

3.5 - User can search for ads by event title/description	✓	This method of searching was used in our Dafny verification so it was given Priority 2 and was quickly implemented
3.6, 3.6.1-2 - Bidder can look at detail of an advertisement before bidding: Poster details, Ad description	✓	This is a very useful tool in our system and was carried out in accordance with its Priority 2 status
3.7 - A bidder can remove a bid from an active advertisement	✓	While we only assigned this Priority 3 despite the importance of this feature in our system, it was implemented earlier anyway

7. Teamwork

Team member	Responsibility
Bailey	<ul style="list-style-type: none"> ● Team leader ● Front end (HTML, CSS, Javascript) ● Back end (Python, flask) ● Database schema, queries and functions ● Initial/final report ● Dafny translation ● WBS ● Demonstration preparation
Harry	<ul style="list-style-type: none"> ● Dafny verification of functions and ArrayList datatype ● Dafny to Python translation ● Back-end verification summary
Nabil	<ul style="list-style-type: none"> ● Front end (HTML, CSS, Javascript) ● Database schema, queries and functions ● Flask login ● Demonstration presentation ● Modals ● User dashboard
Estella	<ul style="list-style-type: none"> ● Front end (HTML, CSS, Javascript) ● Initial/final report ● Testing
Jacob	<ul style="list-style-type: none"> ● Front end (HTML, CSS, Javascript) ● Initial/final report ● Black-Box Testing
All members	<ul style="list-style-type: none"> ● Requirements

8. Appendices

References:

[1] - <https://www.microsoft.com/en-us/research/wp-content/uploads/2016/12/krml203.pdf>

Verification Sources:

<https://ieeexplore.ieee.org/document/8106874/>
<http://orbit.dtu.dk/files/51244038/DafnyLectureNotes.pdf>
<https://ece.uwaterloo.ca/~agurfink/stqam/2018/03/03/dafny-ref>
<https://eprints.ucm.es/38702/1/memoria.pdf>
<https://www.cs.cmu.edu/~mfredrik/15414/lectures/17-notes.pdf>
https://www.doc.ic.ac.uk/~scd/Dafny_Material/Lectures.pdf
<https://www.engr.mun.ca/~theo/Courses/AlgCoCo/6892-downloads/dafny-notes-010.pdf>
<http://dafny2.rssing.com/chan-33926459/latest.php>
https://books.google.com.au/books?id=t0G5BQAAQBAJ&pg=PA391&lpg=PA391&dq=filter+dafny&source=bl&ots=V8zfH7IRJK&sig=dGkr9MirARVFn_i7huj-fpJ_Xo&hl=en&sa=X&ved=2ahUKEwiJ5u6bkaTeAhVGQo8KHeTXBVQQ6AEwBnoECAAQ#v=onepage&q=filter%20dafny&f=false
<https://www.youtube.com/watch?v=fSWZWXX5ixc>
<https://www.youtube.com/watch?v=dQC5m-GZYbk>
<https://www.cs.bgu.ac.il/~ccpr171/wiki.files/Substring.dfy>
<https://www.scss.tcd.ie/Vasileios.Koutavas/teaching/cs4004-4504/mt1718/project-part-1/>
<https://stackoverflow.com/questions/41397455/dafny-substring-implementation>
<http://www.cs.rpi.edu/academics/courses/spring18/csci2600/Lecture9.pdf>
<http://soe.rutgers.edu/sites/default/files/imce/pdfs/gset-2014/Formal%20Verification%20with%20Dafny.pdf>